Computer Science

Writing Algorithms & Pseudo-code

(Top Tips for Improving Exam Technique)

Introduction

In my experience, students who are otherwise very competent and able programmers, often come unstuck when confronted with questions asking them to write code / algorithms under exam conditions.

The purpose of this booklet is to provide you with a few tips to ensure you gain the highest possible marks for these types of question... because you will have to face them sooner or later!

Top Tips for Coding Questions

1. **Don't panic!**
2. Read the question thoroughly
3. Breathe...
4. Read the question again
5. Underline / highlight key points in the question
6. Sketch a flow diagram / map out the logic on the question paper
   (Use any blank space on the paper – cross out when finished)
7. Attempt the question in a cool, calm & logical manner

Reading the Question

The most important part of the process is to read the question thoroughly to ensure you understand exactly what it is asking you to do. It is essential that you know whether you are writing ***pseudo-code*** or ***actual code***! You can expect the following types of question:

1. Scenario provided, you are asked to write an algorithm or pseudo-code

- Stick to pseudo-code

e.g. *Scenario....... Write an* ***algorithm*** *for this process.*

2. Scenario provided, you are asked to write proper code

- Use full code in your answer
- Do not mix with pseudo-code
- State the name of the language being used
- Use indentations, meaningful variable names, comments etc...

e.g. *Write the code for the subroutine .... in a* ***high level language****. You should* ***state the name of the language*** *you have used and use* ***good program writing techniques*** *to ensure that your code can be understood by another programmer.*

3. Code provided, you are asked to adapt / modify existing code

- Use the same format as the example provided – this may be a generic style of coding or pseudo-code.

e.g. *The algorithm for updating points in the case of a draw is given below.*

```
IF goals_of_first_team = goals_of_second_team THEN
     points_of(first_team) = points_of(first_team) + 1
     points_of(second_team) = points_of(second_team) + 1
END IF
```

Using the same format*, write the* ***algorithm*** *for updating the points if there is a winner.*

Let's take a look at an example, based on a question from the June '09 paper...

A student has written a program which stores and prints recipes. The program can be used to calculate the quantities necessary to cater for different numbers of people.

- The user inputs the new number of people
- The data for the recipe is read one line at a time
- For each ingredient the new quantity is calculate
- The output is written to a new file

Write an algorithm for this process ***[8 marks]***

Firstly, let's highlight the important bits...

A student has written a program which stores and prints recipes. The program can be used to calculate the quantities necessary to cater for different numbers of people.

- The user inputs the new number of people
- The data for the recipe is read one line at a time
- For each ingredient the new quantity is calculated
- The output is written to a new file

Write an algorithm for this process ***[8 marks]***

What can we establish from the question?

Write your answer in pseudo-code (no specific mention of a high level language)

Algorithm must accept INPUT from user (number of people)

Algorithm must OPEN a file to READ recipe information

Algorithm must OPEN a NEW file to WRITE new recipe information

Algorithm must use a LOOP to:

Read each item in file one by one

Calculate new quantity for each item

WRITE item and new quantity to new file

Algorithm must CLOSE file being WRITTEN to

Algorithm must CLOSE file being READ from

Convert this into pseudo-code...

```
numPeople = INPUT(number of people)          // 1 mark - user input

File Open to READ (recipeFile)               // 1 mark - file open (read)

File Open to WRITE (outputFile)              // 1 mark - file open (write)

For (each item in recipeFile)                // 1 mark - LOOP all items in file

    Read ingredient & quantity               // 1 mark – read each line

    newQuantity = quantity * numPeople       // 1 mark – calculate new quantity

      Write ingredient to OutputFile         // 1 mark – write item to new file

      Write new Quantity to OutputFile

Next

File Close (recipeFile)                      // 1 mark for closing all files

File Close (outputFile)
```

Easy, isn't it?

Now it's your turn. Work through the following examples. Use the techniques described to extract all of the useful information from the question. Be calm. Think logically.

You will find the complete mark schemes for each question at the end of the paper. It is for your own benefit to attempt the questions before looking at the answers!

1. Attempt the question to the best of your ability
2. Check your answers against the mark scheme
3. Try the question again
4. Go on to the next question

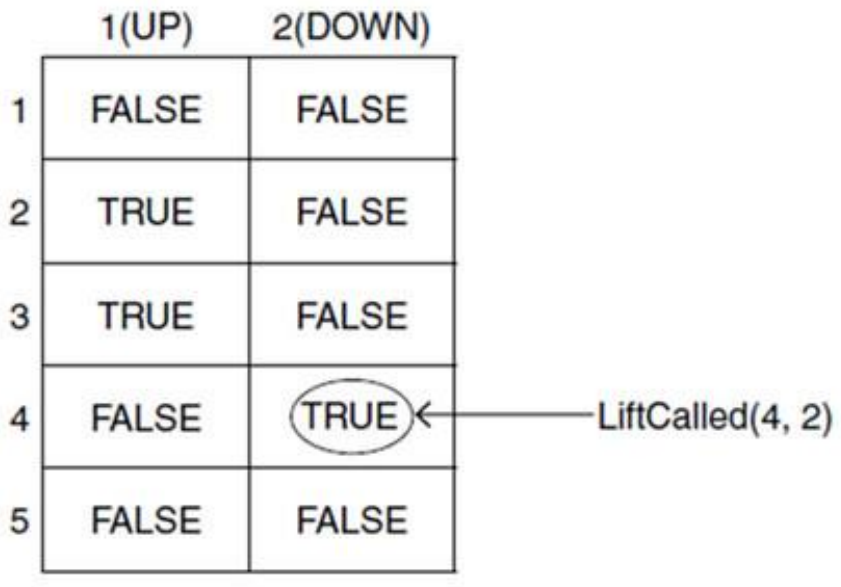
Question 1

Paper	F452	Question	3 (d)
Year	June 2009	Marks Available	6

Scenario

A company is writing a program to control the lift in a 5-storey building. The program uses an array called LiftCalled( ) to store whether the lift has been called on each floor.

The structure of the array used is shown in the table below. Each row represents a floor of the building. The first column stores whether a lift is wanted to go UP from that floor, and the second column stores whether the lift is wanted to go DOWN from that floor.



	1(UP)	2(DOWN)
1	FALSE	FALSE
2	TRUE	FALSE
3	TRUE	FALSE
4	FALSE	TRUE
5	FALSE	FALSE

LiftCalled(4, 2)

Question

Supervisors can call the lift using an override facility. This uses a subroutine call SupervisorCall. This subroutine has one parameter, Floor (the number of the floor where the supervisor is.) The subroutine sets the UP and DOWN values for the designated floor to TRUE, and all other values in the array to FALSE.

Write the code for the subroutine SupervisorCall in a high level language. You should state the name of the language you have used and use good program writing techniques to ensure that your code can be understood by another programmer.

Name of Language ....

Code

....

....

....

....

....

....

....

....

....

....

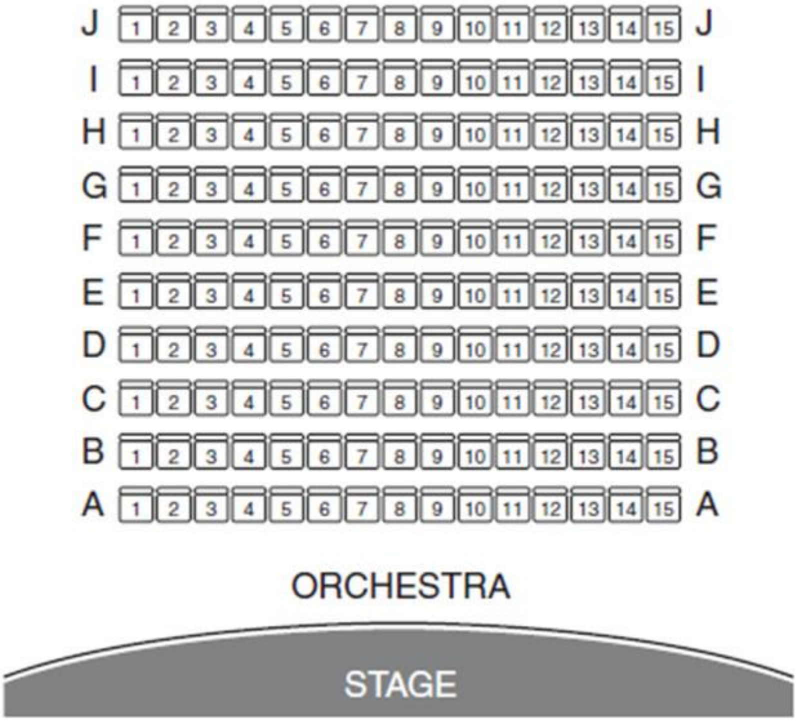
....

.... **[6]**

Question 2

Paper	F452	**Question**	4 (c)
Year	June 2009	**Marks Available**	8

A programmer is producing a computer program which allocates seats to customers in a small theatre. The theatre has 10 rows, labelled A to J from front to back, and 15 seats in each row, numbered 1 to 15 from left to right.



The program uses the following rules to choose the best seats.

- **Rule 1: All seats in one booking must be in the same row, next to each other.**
- **Rule 2: The seats must be as close to the front as possible.**

Write an algorithm which takes the number of tickets wanted as an input, and outputs the best seats available.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[8]**

Question 3

Paper	F452	**Question**	4 (e)
Year	January 2010	**Marks Available**	7

Scenario

A mail order company charges for delivery depending on the volume and the weight of the items purchased. A computer program processes orders and calculates the cost of delivery.

The code for calculating the total weight and the total volume of the items purchased is shown below.

```
TotalWeight = 0
TotalVolume = 0

FOR i = 1 TO NumberOfItemsOrdered
  TotalWeight = TotalWeight + WeightOfItem(i)
  TotalVolume = TotalVolume + VolumeOfItem(i)
NEXT i
```

Question

The cost of delivery is calculated as follows:

- There is a basic delivery charge of £5 for all orders.
- If the total weight of an order is more than 1 kg, there is an additional charge of £0.50 for every extra 0.1 kg.
- If the total volume of an order is more than 1000 cm3, there is an additional charge of £0.50 for every extra 200 cm3.

Write an algorithm for a function which makes use of the global variables TotalWeight and TotalVolume and returns the cost of delivery.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[7]**

Question 4

Paper	F452	Question	1 (b)
Year	June 2010	Marks Available	4

Scenario

A computer program is designed to store the results of matches in a football competition and calculate the ranking of the teams. To enter the results of each match, an interface will allow the user to select the name of two teams from drop down lists, and enter the number of goals scored in corresponding text boxes.

Question

When the results are entered, the number of points of each team is updated as follows:

- If both teams have the same number of goals (draw) then each team gets 1 point.
- If one team has more goals than the other (i.e. there is a winner) then the winning team gets 3 points, and the losing team gets 0 points.

The algorithm for updating points in the case of a draw is given below.

```
IF goals_of_first_team = goals_of_second_team THEN
    points_of(first_team) = points_of(first_team) + 1
    points_of(second_team) = points_of(second_team) + 1
END IF
```

Using the same format, write the algorithm for updating the points if there is a winner.

....

....

....

....

....

.... **[4]**

Question 5

Paper	F452	**Question**	2 (f)
Year	June 2010	**Marks Available**	5

The playing length of each song to be written to a CD is stored in an array called SongLength. When the user wants to write the songs selected onto a CD, the software must check that the total playing length does not exceed 80 minutes.

The software contains the following function to perform this check.

```
01 FUNCTION CheckTotalLength() : BOOLEAN
02
03   TotalLength = 0
04
05   FOR i = 1 TO NumberOfSongs
06     TotalLength = TotalLength + SongLength(i)
07   NEXT i
08
09   RETURN (TotalLength > 80)
10
11 END FUNCTION
```

The algorithm for the function CheckTotalLength() contains a FOR loop in lines 05 to 07. Rewrite this FOR loop as a WHILE loop.

....

....

....

....

....

....

....

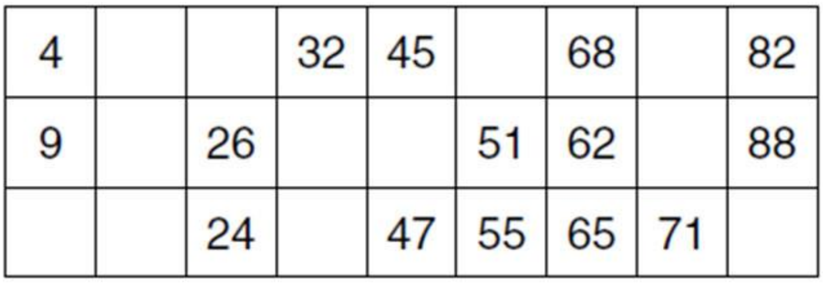
.... **[5]**

Question 6

Paper	F452	**Question**	3 (c)
Year	June 2010	**Marks Available**	8

Scenario

A printing company uses a computer program to randomly generate and print bingo tickets. Each bingo ticket has a grid with three rows and nine columns. Each row contains 5 numbers and 4 blank spaces. The computer program stores the numbers in a 2-dimensional array called Ticket.



4			32	45		68		82
9		26			51	62		88
		24		47	55	65	71	

After filling the array, the computer program ensures that no numbers have been repeated, and replaces four positions on each row with the number 0.

Question

The bingo ticket is then printed using the following method.

- For every row in the array
- For every column in that row
    - If the value is 0 then output a space,
    - otherwise output the value

Write an algorithm in pseudo-code to print the numbers in the array onto a ticket.
You should indent your pseudo-code correctly to make it easier to understand.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[8]**

Trace Table Questions

You will occasionally come across questions that provide you with an algorithm and require you to trace the values of variables during its execution. Again, these questions are very easy if you keep calm and think logically! Just take them one line at a time & make sure you understand which values you should be tracing!

Question 7

Paper	F452	**Question**	4 (d)
Year	June 2010	**Marks Available**	5

Scenario

Numerology is a method of fortune-telling where letters are converted into numbers.
A programmer is writing an application to carry out this conversion.

The application contains the following function:

```
FUNCTION Mystery(n : Integer) : Integer

  IF n < 10 THEN
       RETURN n
  ELSE
       RETURN Mystery (n - 9)
  END IF

END FUNCTION
```

Question

Trace the execution of the call Mystery(15), showing every function call and the value returned.

(You may use a diagram.)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[5]**

Question 8

Paper	F452	**Question**	4 (d)
Year	January 2010	**Marks Available**	9

Question

The code for calculating the total weight and the total volume of the items purchased is shown below.

```
10 TotalWeight = 0
11 TotalVolume = 0
12
13 FOR i = 1 TO NumberOfItemsOrdered
14   TotalWeight = TotalWeight + WeightOfItem(i)
15   TotalVolume = TotalVolume + VolumeOfItem(i)
16 NEXT i
```

A dry run uses the following test data:

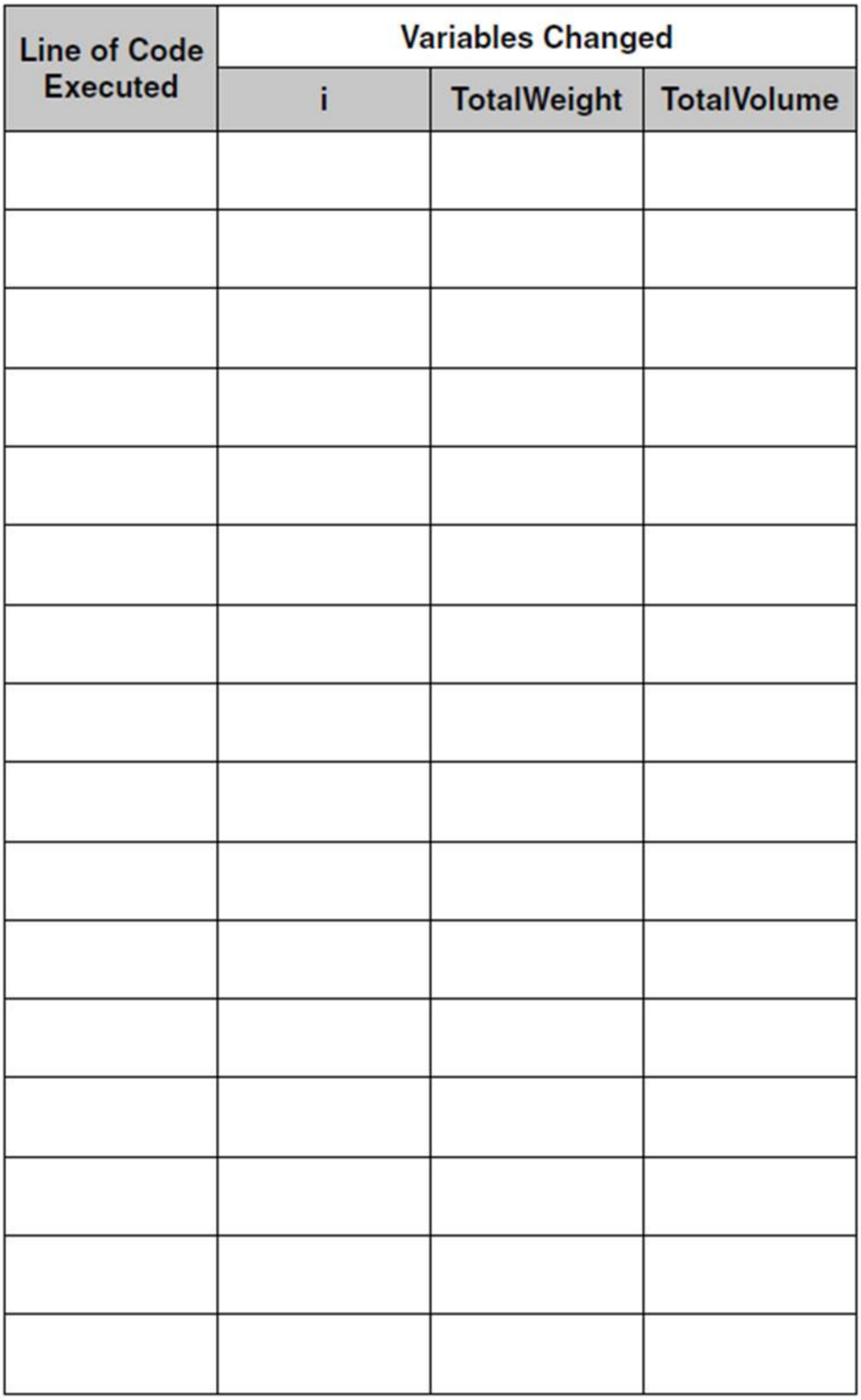
NumberOfItemsOrdered = 2

WeightOfItem(1) = 0.3
VolumeOfItem(1) = 200

WeightOfItem(2) = 0.1
VolumeOfItem(2) = 150

Complete the trace table opposite, showing each line of the algorithm which will be executed. On each line, write down the new values of any variables that are changed.

You may not need every row in the table.



Line of Code Executed	Variables Changed		
	i	TotalWeight	TotalVolume

[9]

Question 9

Paper	F452	**Question**	5 (e)
Year	January 2009	**Marks Available**	6

Scenario

A coin-operated vending machine has 2 slots. Slot A is for 10p coins, and slot B is for 5p coins. The software of the machine has a function called CoinValue which takes a single character "A" or "B" and returns the value of a coin which has dropped in that slot as an integer.

The machine records the coins that have been entered using a string of the characters A and B. (So "ABB" means that a 10p coin was entered followed by two 5p coins.)

The software in the machine uses the following recursive function:

```
01 BEGIN Function Calculate(CoinString)
02
03   IF Length of CoinString = 1 THEN
04     Calculate = CoinValue(CoinString)
05   ELSE
06     First = First Character in CoinString
07     Rest = All the characters in CoinString after the first
08     Calculate = CoinValue(First) + Calculate(Rest)
09   END IF
10
11 END FUNCTION
```

Question

The function Calculate is called with the argument "AB".

Trace the execution of Calculate("AB") indicating clearly

- Each time the function is called
- The value of the argument in each call
- The lines of the algorithm that are executed
- The value that is returned from each function call

You may use a diagram in your answer.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

[6]

Mark Scheme

Question 1

Example:

```
Sub SupervisorCall(Floor : Integer)

  dim iFloor As Integer
  dim iDirection As Integer

  FOR iFloor = 1 to 5
    FOR iDirection = 1 to 2
      IF iFloor = Floor
        LiftCalled(iFloor, iDirection) = TRUE
      ELSE
        LiftCalled(iFloor, iDirection) = FALSE
      END IF
    NEXT
  NEXT

END SUB
```

Mark points for:

- Using "SupervisorCall" as identifier of subroutine
- Using 1 Parameter called Floor (of data type Integer)

[2]

- Setting all values of array LiftCalled to False...
- ... except for LiftCalled(Floor, 1) and LiftCalled(Floor,2) which are set to TRUE

[2]

- Correct use of indentation
- Code annotated / can be understood easily without comments
- Descriptive identifier names

[Max 2 for above]

Question 2

1 Mark per bullet point (to a maximum of 8):

- Input NumberOfTickets, n
- If n < 0 or n > 15 produce an error message
- And stop

(Else)

- Loop through the rows from Row A
- Until seats are found
- ... or you reach row J

(provided this is within a loop)

- Test that there are n seats available in row (together)
- .... Eg by finding first empty seat
- ..... and checking n – 1 seats after it.

- If seats found then output the seat numbers.
- Else output appropriate message.

[8]

Question 3

Any 7 of the following mark points (irrespective of method used to present the algorithm):

- An algorithm for a function which returns the cost of delivery (accept incorrect calculations, provided the function returns the result of the calculations)
- The variables *TotalWeight* and *TotalVolume* are used in the calculations (as global variables or as parameters of the function)
- Determines amount of excess weight if any
- Determines cost of excess weight
- Determines amount of excess volume if any
- Determines cost of excess volume
- Correctly calculates cost of delivery

Example:

```
FUNCTION CostOfDelivery()

  IF TotalWeight > 1 THEN
    ExtraWeight = TotalWeight – 1
    ExtraWeightUnits = ExtraWeight DIV 0.1
    ExtraWeightCost = ExtraWeightUnits * 0.5
  ELSE
    ExtraWeightCost = 0
  END IF
  IF TotalVolume > 1000 THEN
    ExtraVolume = TotalVolume – 1000
    ExtraVolumeUnits = ExtraVolume DIV 200
    ExtraVolumeCost = ExtraVolumeUnits * 0.5
  ELSE
    ExtraVolumeCost = 0
  END IF

  Cost of delivery = 5 + ExtraVolumeCost + ExtraWeightCost

END FUNCTION
```

[7]

Question 4

Example:

```
IF goals_of_first_team > goals_of_second_team THEN
      points_of(first_team) = points_of(first_team) + 3
ELSE IF goals_of_second_team > goals_of_first_team THEN
      points_of(second_team) = points_of(second_team) + 3
END IF
```

Mark points for:

- Correctly, compares the goals scored by both teams (> or <)
- 3 points added to first team if it is the winner
- 3 points added to second team if it is the winner
- Correct nesting/ use of ELSE IF statements

[1 per point, max 4]

Question 5

Example:

```
i = 1

WHILE i <= NumberOfSongs
  TotalLength = TotalLength + SongLength(i)
  i = i + 1
END WHILE
```

Several other solutions are possible.

Award marks for the following mark points:

- i initialised correctly (typically 1 or 0)
- WHILE loop has been used correctly...
- ... and the condition of the while loop (and the initial value of i) ensure that the correct number of iterations are made
- Songlength(i) is added to the TotalLength within each iteration
- The value of i is incremented (or decremented, as appropriate) within each iteration

[1 per point, max 5]

Question 6

Example:

```
FOR Row = 1 TO 3

  FOR Column = 1 TO 9
    IF Ticket(Row, Column) = 0 THEN
      PRINT " "
    ELSE
      PRINT Ticket(Row, Column)
    END IF
  NEXT Column

  Go to new line

NEXT ROW
```

1 Mark per bullet point (to a maximum of 8):

- Correct row loop
- Correct column loop
- Correctly nested/considers every element of the array Ticket
- Checks if the value is 0 and ...
- ... prints a space if it is 0
- ... prints the value if it is not 0
- There is a new line at the end of each row
- Correct indentation has been used

[8]

Question 7

1 Mark per bullet point (to a maximum of 5):

Call Mystery(15) / n = 15 **[No mark for this]**

- Line 53 – condition is FALSE
- RETURN Mystery(15 – 9) (in line 56)
- ... New call to Mystery(6)/n = 6
- ... ... Line 53 is TRUE
- ... ... Return 6 to Mystery (15 - 9) (in line 56 of previous call)
- Final return value is 6

Candidates do not need to refer to line number to have the mark (but it must be clear what line of code they are on)

[5]

Question 8

1 mark per correct value, changed on the correct line

Line of Code Executed	Variables Changed		
	i	TotalWeight	TotalVolume
10		0	
11			0
13	1		
14		0.3	
15			200
16	2		
13			
14		0.4	
15			350
16	3		
13			

[9]

Question 9

Award 1 mark per bullet to a maximum of 6:

- Function called with argument (“AB”)
- Line 2 IF statement false so do lines 4/5
- Line 7 CoinValue(A) = 10
- Line 7 needs Calculate (“B”) – so new call
    - Function called with argument (“B”)
    - Line 2 IF Statement is TRUE so do line 3
    - Calculate = CoinValue(“B”) = 5
    - Call ends returning 5
- Line 7 continues: Return value = 10 + 5 = 15
- Call ends returning 15

[6]

Your Scores

You can use this table to keep track of your progress and scores...

Question	Maximum	Attempt 1	Attempt 2
1	6		
2	8		
3	7		
4	4		
5	5		
6	8		
7	5		
8	9		
9	6		
Total	**58**		